

APPENDIX A

```
/*
 * Domain.java
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 */
package com.genmagic.util;

public interface Domain {
    public boolean isVisited();
    public boolean setVisited(boolean b);
    public String getName();
}
```

```

/*
 * DomainController.java
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 */
package com.genmagic.util;

import java.util.Vector;

public class DomainController implements java.io.Serializable {
    Vector _domains = new Vector();
    int _current = 0;

    public DomainController() {
    }

    public void setDomains(Vector v) {
        _domains = v;
    }

    private Domain getCurrentDomain() {
        return getDomainAt(_current);
    }

    private Domain getDomainAt(int i) {
        return (Domain)_domains.elementAt(i);
    }

```

```

}

public boolean isCurrentDomainVisited() {
    return getCurrentDomain().isVisited();
}

public String getDomainName() {
    return getCurrentDomain().getName();
}

public void moveToNext() {
    for (;_current < _domains.size()-1; _current ++ ) {
        if (getCurrentDomain().isVisited() == false) {
            getCurrentDomain().setVisited(true);
            return;
        }
    }
}

public void moveToPrevious() {
    _current --;
}

public boolean hasPrevious() {
    return _current >0;
}

public boolean hasMore() {
    for (int i=_current; i < _domains.size()-1; i ++ ) {
        if (getCurrentDomainAt(i).isVisited() == false) {
            return true;
        }
    }
}

```

```

    }
    }
    return false;
}
}

```

[illegible]

```
<?xml version="1.0"?>
<!--
 * FILE_NAME
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
-->

<% String DNIS = request.getParameter("DNIS");
   String ANI = request.getParameter("ANI"); %>
<vxml version="1.0">

  <form>
    <block>
      <% if (DNIS.startsWith("800123")) { %>
        <!-- rout a call to banking service number to banking application -->
        <goto next="http://banking.genmagic.com/bankingService.jsp"/>
      <% if (ANI.startsWith("8004567890")) { %>
        <!-- rout a call from specific number to specific URL -->
        <goto next="http://my.genmagic.com/oshima.jsp"/>
      <% } %>
    </block>
  </form>
</vxml>
```



```
public String getPrompts() {
    return toVXML();
}

abstract protected Vector construct();

protected static java.util.Random rand = new java.util.Random();

public String toVXML() {
    if (_cache == null) {
        synchronized (this) {
            if (_cache == null) {
                _cache = construct();
            }
        }
    }
    return (String)_cache.elementAt( (int)(rand.nextFloat() * _cache.size()));
}
}
```

```

/*
 * Message.java
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 */
package com.genmagic.util;

import java.util.Date;

public interface Message {
    public String getForm();
    public String getSubject();
    public Date getDate();
    public String getBody();
}

```



```
/*
 * EmailList.java
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 */
package com.genmagic.util;

import java.util.Vector;
import java.util.Enumeration;

public class EmailList implements java.io.Serializable {
    Vector _messages = new Vector();
    int _current = 0;
    public EmailList() {
    }

    public void setMessages(Vector v) {
        _messages = v;
    }

    public Message getCurrentMessage() {
        return (Message)_messages.elementAt(_current);
    }

    public int getNumberOfMessage() {
        return _messages.size();
    }
}
```



```

/*
 * RandomPrompts.java
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 */
package com.genmagic.util;

import java.util.Vector;
import java.util.Enumeration;

public class RandomPrompts extends PromptSet {

    synchronized protected Vector construct() {
        int size = _prompts.size();
        Vector cache = new Vector();

        int base = 0;

        int six  = size % 6;
        int five = size % 5;
        int four = size % 4;

        int min = Math.min(six, five);
        int max = Math.max(six, five);

        if (size <= 8) {

```



```

buf.append("<var name='tmp' expr='Math.random()*" +
          n + "'/>\n");
buf.append("<if cond='1.0>=tmp'>\n");
buf.append("<audio src='builtin:" + _prompts.elementAt(i) + "'/>\n");
i++;
for (int j=1; j < n && i < size; j++, i++) {
    buf.append("<elseif cond='" + (j+1) + ".0>=tmp' />\n");
    buf.append("<audio src='builtin:" + _prompts.elementAt(i) + "'/>\n");
}
buf.append("</if>");
cache.addElement(buf.toString());
}
if (cache.size()==0) {
    cache.addElement("");
}
return cache;
}

public static void main(String a[]) throws Exception {
    RandomPrompts r = new RandomPrompts();
    r.setAddPrompt("a.wav");
    r.setAddPrompt("b.wav");
    r.setAddPrompt("c.wav");
    r.setAddPrompt("d.wav");
    r.setAddPrompt("e.wav");
    r.setAddPrompt("f.wav");
    r.setAddPrompt("g.wav");

    r.toVXML();
    System.out.println(r.toVXML());
}

```

$$\overbrace{\quad}$$

Parameter	Value	Parameter	Value
α_1	0.0000	α_2	0.0000
α_3	0.0000	α_4	0.0000
α_5	0.0000	α_6	0.0000
α_7	0.0000	α_8	0.0000
α_9	0.0000	α_{10}	0.0000
α_{11}	0.0000	α_{12}	0.0000
α_{13}	0.0000	α_{14}	0.0000
α_{15}	0.0000	α_{16}	0.0000
α_{17}	0.0000	α_{18}	0.0000
α_{19}	0.0000	α_{20}	0.0000
α_{21}	0.0000	α_{22}	0.0000
α_{23}	0.0000	α_{24}	0.0000
α_{25}	0.0000	α_{26}	0.0000
α_{27}	0.0000	α_{28}	0.0000
α_{29}	0.0000	α_{30}	0.0000
α_{31}	0.0000	α_{32}	0.0000
α_{33}	0.0000	α_{34}	0.0000
α_{35}	0.0000	α_{36}	0.0000
α_{37}	0.0000	α_{38}	0.0000
α_{39}	0.0000	α_{40}	0.0000
α_{41}	0.0000	α_{42}	0.0000
α_{43}	0.0000	α_{44}	0.0000
α_{45}	0.0000	α_{46}	0.0000
α_{47}	0.0000	α_{48}	0.0000
α_{49}	0.0000	α_{50}	0.0000
α_{51}	0.0000	α_{52}	0.0000
α_{53}	0.0000	α_{54}	0.0000
α_{55}	0.0000	α_{56}	0.0000
α_{57}	0.0000	α_{58}	0.0000
α_{59}	0.0000	α_{60}	0.0000
α_{61}	0.0000	α_{62}	0.0000
α_{63}	0.0000	α_{64}	0.0000
α_{65}	0.0000	α_{66}	0.0000
α_{67}	0.0000	α_{68}	0.0000
α_{69}	0.0000	α_{70}	0.0000
α_{71}	0.0000	α_{72}	0.0000
α_{73}	0.0000	α_{74}	0.0000
α_{75}	0.0000	α_{76}	0.0000
α_{77}	0.0000	α_{78}	0.0000
α_{79}	0.0000	α_{80}	0.0000
α_{81}	0.0000	α_{82}	0.0000
α_{83}	0.0000	α_{84}	0.0000
α_{85}	0.0000	α_{86}	0.0000
α_{87}	0.0000	α_{88}	0.0000
α_{89}	0.0000	α_{90}	0.0000
α_{91}	0.0000	α_{92}	0.0000
α_{93}	0.0000	α_{94}	0.0000
α_{95}	0.0000	α_{96}	0.0000
α_{97}	0.0000	α_{98}	0.0000
α_{99}	0.0000	α_{100}	0.0000